

RESEARCH ARTICLE

Open Access

Social BIMCloud: a distributed cloud-based BIM platform for object-based lifecycle information exchange

Moumita Das^{*}, Jack CP Cheng^{*} and Srinath S Kumar

Abstract

Background: The architecture, engineering and construction (AEC) industry lacks a framework for capturing, managing, and exchanging project, product, and social information over the lifecycle of a building. The current tools have various limitations, such as lack of interoperability, slow to transfer huge building model files, and possibility of data inconsistency.

Methods: In this paper, we present a cloud-based BIM server framework namely Social BIMCloud that facilitates BIM information exchange through dynamic merging and splitting of building models. The data model of Social BIMCloud is based on but not limited to IFC. The data model of Social BIMCloud was further extended to accommodate social interactions, by studying the formal modes of communication in the AEC industry. An object-based approach to capture and manage social interactions in AEC projects through a BIM-based visual user interface was also developed and demonstrated.

Results: Social BIMCloud addresses the issues of inefficient data transfer speed and data inconsistency in a distributed environment by facilitating the storage and partial exchange of integrated nD BIM models. Data interoperability is facilitated through open BIM standards such as IFC and direct integration with construction software. High performance, scalability, fault tolerance, and cost effectiveness are facilitated through data partitioning, data replication strategies, multi-node structures, and pay-per-use tariff systems, respectively, through a cloud-based NoSQL database.

Conclusion: The Social BIMCloud framework helps to develop and exchange BIM models, which are rich in project information such as social interactions, cost, and energy analyses. This framework improves the communication efficiency between project participants, leading to better designs and less rework. The information captured by this framework could also be useful to determine important metrics such as industry trends, relationships among project participants, and user requirements.

Keywords: Building information modeling; Cloud computing; Distributed data storage; Partial information exchange; Social interaction

^{*} Correspondence: mdas@connect.ust.hk; cejcheng@ust.hk
Department of Civil and Environmental Engineering, The Hong Kong
University of Science and Technology, Clear Water Bay, Kowloon, Hong
Kong, China

Background

Building information modeling (BIM) is increasingly being adopted in the architecture, engineering, and construction (AEC) industry to represent, store, share, and manage building information. However, huge size of BIM files, multi-domain fragmented nature of the industry, and file-based exchange of BIM information lead to problems like data transfer inefficiency, lack of interoperability, and data inconsistency. Data transfer efficiency is hindered as BIM files are usually large in size and are required to be transferred to remote construction sites with limited Internet bandwidth. However, data transfer efficiency can be improved by exchanging partial BIM files which are smaller in size and contain BIM information relevant to a specific project partner. Research has been done to facilitate splitting of BIM models for generating partial BIM files and multi-model views (Katranchukov et al. 2003; Nepal et al. 2009; Redmond and Smith 2011; Nour 2007). For example, Redmond and Smith (2011) used simple XML formats, called SML (Simplified Markup Language) for representing and exchanging partial geometric information. The SML, which represents only the geometry of a BIM model, is devoid of semantic information, and therefore cannot be considered as true BIM. BIMServer (Beetz et al. 2010) facilitates partial updating and retrieval of BIM files only through partitions decided at the beginning of the project, and therefore does not facilitate dynamic splitting and merging of BIM models.

Apart from improving data transfer efficiency, partial information exchange also ensures data consistency through server based BIM systems. Project partners from multiple domains such as designers, contactors, and domain specialists, who may be geographically dispersed, work in parallel on different parts of the same building model during the design phase. Through dynamic partial information exchange on a server based BIM model, all the project partners can retrieve and update real time information from their own parts and the rest of the common integrated BIM model. In addition, by working on an integrated BIM model, a project partner may be aware of the changes made in the same or different domains of the BIM model by other project partners. This can ensure that project partners have the same piece of information, thereby facilitating collaborative decision making. In research, server based BIM has been explored to facilitate e-procurement, project planning and design, by using BIM files generated through proprietary BIM software (Shen et al. 2012; Grilo and Jardim-Goncalves 2011; Chuang et al. 2011). However, BIM files generated using proprietary software cannot be split into sub-models if an API (Application Programming Interface) is not provided by the respective organization. This issue can be addressed by open BIM

standards like IFC (Industry Foundation Classes). Server based frameworks have been proposed by researchers that facilitate file-based information exchange of the whole IFC model (Chen et al. 2005; Chan and Leung 2004; Tanyer and Aouad 2005; Plume and Mitchell 2007; Faraj et al. 2000). File-based information exchange does not consider splitting of models and therefore does not improve the efficiency of data transfer. File-based exchange in a fragmented environment also causes data duplicity and hence inconsistency. Therefore, researchers have developed IFC based BIM servers using web based object-relational databases that considers partial information exchange (Nour 2007; Kang and Lee 2009; Nour and Karl 2008). These frameworks however have limitations on the size of data that can be exchanged, as relational databases may not be able to store and exchange large amounts of data in a web based environment. The server based BIM platform called BIM-Server (Beetz et al. 2010) handles large amounts of data but does not facilitate dynamic partial information exchange as described earlier. Clear methods for extending the schema of BIMServer and its integration with external software are also lacking.

Furthermore, the existing frameworks do not consider the capture and management of social interactions, which directly induce design changes in the BIM model. The built environment goes through several changes over its lifecycle, starting from the design phase through the construction phase and finally to the operation and maintenance phase. Especially in the design and construction phases, the original building design may change significantly in terms of placement, material, and use through social interactions among project partners like new user requirements, change orders, and RFIs (Request for Information). These social interactions may contain quality information related to user requirements, changes in schedule due to rework, and cost negotiations, and therefore should be captured and managed efficiently (Nawaz et al. 2012). Knowledge captured from social interactions can be used for detecting communities and community leaders in an organization (Tyler et al. 2003; Huberman and Adami 2004), discovering irregularities in work practice and accident prediction (Nawaz et al. 2012). Moreover, current approaches to BIM servers do not consider the integration of BIM models with lifecycle processes such as energy simulation and construction site layout planning. The AEC industry thus needs a collaborative framework that could facilitate integration of BIM models with lifecycle processes and social interaction records, either formal or informal, throughout a construction project. This aligns with the concept of "Social BIG BIM", which describes a collaborative approach to exchange nD BIM models (for example, phasing-4D, cost-5D, energy performance-6D,

and facility management-7D) with project partners for lifecycle processes (Jernigan 2008). The term “social” has also been used to refer to end user participation in a construction process (Jäväjä et al. 2012). However, this paper focuses on storing and managing an integrated nD BIM model with social interaction information in a distributed environment through cloud-based implementation.

Cost is a very important factor in construction projects, in terms of project cost (Dainty et al. 2001), cost of IT system implementation (Sargent et al. 2012), or cost of BIM adoption in the AEC industry (Liu et al. 2010). Therefore, infrastructure for BIM information exchange should be cost-effective. Cloud computing is a cost-effective and highly scalable means to deliver IT resources and functionality. Cloud computing is a technology that facilitates access to computing applications and resources as services via the Internet on a pay-per-use basis (Mell and Grance 2011). According to a series of interviews with AEC industry professionals conducted by Redmond and Smith (2011), web-based BIM exchanges on cloud platforms can lead to enhanced interoperability between different construction applications. Application of cloud computing technology in the AEC industry has been increasingly studied. For example, Kumar et al. (2010) proposed a cloud-based model for supply chain management. Fathi et al. (2012) proposed a cloud computing-based framework for sharing project information. There are also attempts to integrate cloud computing with BIM. Commercial cloud-based platforms like BIM9 (BIM9 2014) and CaddForce (CaddForce 2014) facilitate file-based information exchange. BIMobject (BIMobject 2014) is a cloud-based application that integrates the native components of BIM software (for example, Autodesk Revit) with actual manufactured products. Other commercial cloud-based software such as Graphisoft BIM Server (Graphisoft 2013) and Autodesk 360 (Autodesk 2013) provide functionalities like querying BIM models as well as graphical interfaces for sharing BIM models in a team. Although several cloud-based BIM platforms exist in the market, the existing cloud-based approaches towards BIM do not provide functionality for capturing non-building related information like social interactions among the design team and the end users. The existing cloud-based BIM platforms also do not facilitate dynamic splitting and merging of BIM models. In addition, those platforms do not allow data replication and connectivity to external programs for extensions of functionality. Furthermore, existing cloud-based platforms such as (Porwal and Hewage 2013; Grilo and Jardim-Goncalves 2011; Jiao et al. 2013) exchange BIM information in a file based manner and therefore hamper data transfer speed and data consistency.

Therefore, in this paper we propose a framework called Social BIMCloud which captures social and

lifecycle information on an integrated BIM model through cloud-based technologies. Social BIMCloud facilitates dynamic splitting and merging of BIM information through open BIM standards like IFC and therefore improves data transfer speed, data consistency, interoperability and cost-effectiveness. Social BIMCloud is based on a NoSQL database management system deployed on a cloud server. A data model was developed with reference to but not limited to IFC schema for storing BIM information in a key-value format in the Social BIMCloud. This data model also captures formal and informal social interactions in a construction project like change orders, RFIs, and user comments. Every property of a building element or social interaction (for example geometry, relation with other elements, and content of interaction) is stored with a unique identifier which is used to facilitate partial updating or retrieval. The data model of Social BIMCloud is also extensible for accommodating data from new domains like energy simulation, cost estimation, and site layout planning. External applications like an energy simulation engine may also be integrated with Social BIMCloud through web service technology. The Social BIMCloud can be integrated with existing BIM software to provide end users with a user-friendly visual user interface for capturing and managing social interactions on different building elements. For demonstrative purpose, Autodesk Revit is currently used for the integration due to its API support and wide use in industry. However, other BIM software can also be integrated with Social BIMCloud without affecting the internal functioning of Social BIMCloud. In addition to this, the Social BIMCloud framework is fault tolerant as it features automatic data replication strategies. It means that the Social BIMCloud framework runs on several cloud machines in parallel and stores copies of BIM information across several nodes (cloud machines situated in different regions around the world), so that the BIM information can be recovered in the case of failure of any node. The Social BIMCloud framework also supports partitioning of big BIM models and stores the partitions on different cloud nodes. This makes the query performance faster by facilitating parallel reads and writes.

Methods

The conceptual system architecture of Social BIMCloud framework

Figure 1 shows the conceptual framework of the proposed Social BIMCloud for storage and exchange of building and social information. The building information refers to the information in a BIM model (for example, building geometry and material information). Social interactions in a construction project may be formal interactions like issuing of change orders and RFIs

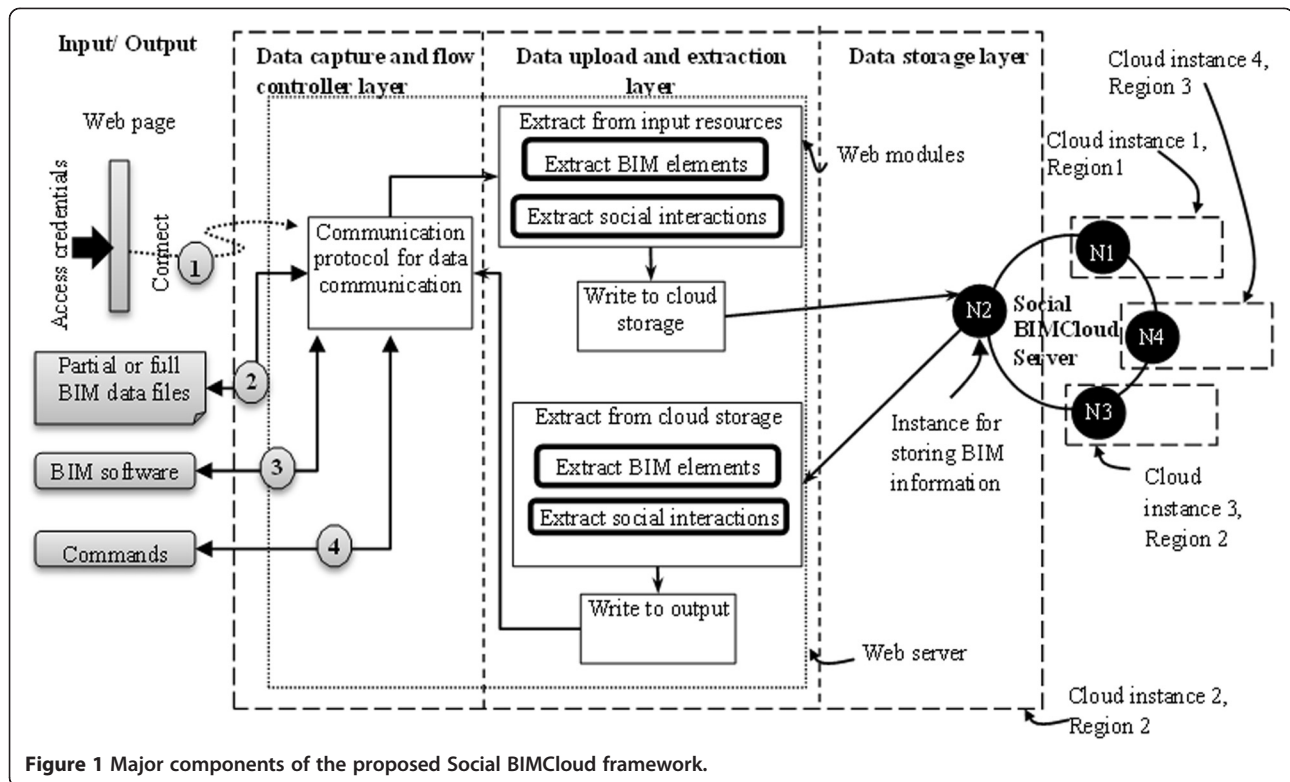


Figure 1 Major components of the proposed Social BIMCloud framework.

or informal interactions such as approving or disapproving a construction process through non-technical comments (as described in 1). The Social BIMCloud framework comprises multiple cloud instances, as shown in Figure 1. A cloud instance refers to a virtual server instance running on a cloud network. A cloud network comprises a large number of computers working in parallel through a real-time communication network like the Internet. Such virtual servers can be formed by dynamically connecting multiple machines (provided by cloud service providers for a fee which is pay-per-use) through the Internet. Therefore the cloud network can be easily scaled up or down for improving the performance, as per the requirement or size of an organization. End users can connect to the Social BIMCloud framework through a front end in two steps (as shown in '1' and '2' in Figure 1). In the first step, end users authenticate their credentials. This front end is a static website hosted on the Social BIMCloud server and acts as a gateway connection to it. In step 2, end users connect to the first layer of the Social BIMCloud after successful authentication (as shown in '2' of Figure 1). In step 2 end users can upload files or execute queries on the Social BIMCloud framework. The proposed Social BIMCloud framework can be divided into three layers – (1) data capture and flow controller layer, (2) data upload and extraction layer, and (3) data storage layer, as explained in Section Data capture and flow controller layer,

Section Data upload and extraction layer, and Section Data storage layer, respectively. Section Scalability of the Social BIMCloud framework and Section Data security on the Social BIMCloud framework describe the scalability and data security of the Social BIMCloud server, respectively. The Social BIMCloud framework uses partial and full building models in the IFC data standard (but is not limited to IFC) for input and output. Visual BIM platforms like Revit, custom designed web pages, or structured queries are used to capture and share BIM and social interactions.

Data capture and flow controller layer

The data capture and flow controller layer takes the input from end users. Based on the input, automated programs hosted on this layer orchestrate the control to the modules of the other layers for information extraction and storage. The design of these program modules has been kept independent and modular. Therefore, modification of one module would not impact the functions of the other. The data capture and flow controller layer comprises a standard data communication protocol, for example, scripting languages like PHP (hypertext pre-processor) and JSP (java server pages). This layer may accept inputs in four methods – (1) through web pages hosted on the Social BIM sever, (2) through BIM files in open standard like IFC, (3) through proprietary BIM software like Autodesk Revit, and (4) through system

Table 1 Examples of types of social interactions captured by Social BIMCloud

| Attribute | Compulsory | Mode of data capture | Examples of user defined values |
|----------------------------|------------|----------------------|---|
| user (issuer) | Yes | Automatic | jcheng, mdas |
| Interaction_key_identifier | Yes | Automatic | |
| Interaction_ID | No | User defined | ID of change order |
| Interaction_type | Yes | User defined | RFI - Substitution, RFI - Clarification, RFI - Deficiency, Change Order - Change in scope, Change Order - Professional errors and omissions, Change Order - Substitution, Change Order - General, Change Order - Design Change, Comment |
| Date_submitted | Yes | Automatic | |
| Response_date | No | User defined | Expected response date, for example, 2 days from issuing of a change order |
| Date_closed | Yes | Automatic | |
| Interaction_status | No | User Defined | Requested, under evaluation, reassess cost, rejected, approved, closed |
| Approval_by_architect | No | User defined | approved, rejected, on hold |
| Approval_by_contractor | | | approved, rejected, on hold |
| Approval_by_owner | No | User defined | approved, rejected, on hold |
| Response_awaiting | No | User defined | owner ID, contractor ID, architect ID |
| Timestamp | Yes | Automatic | |
| Document_URL | No | User defined | PDF document of change order |

commands via secure connections like SSH (Secure Shell). As the storage in the Social BIMCloud is object-based, it accepts both full and partial open BIM standard models. BIM software like Autodesk Revit provides API using which customized plugins can be developed and integrated with it. Using such plugin, end users can directly connect to the Social BIMCloud from their copy of Revit. Customized web pages hosted on the Social BIMCloud can also be devised to take information from end users through web-based forms. By using NoSQL database query languages like CQL (Cassandra Query Language developed to query the NoSQL database, Apache Cassandra), end users can also execute queries to update or retrieve customized information from the data storage layer through secure connections. CQL is based on SQL (Structured Query Language) which is the standard for relational database manipulation. The syntax of CQL for updating and retrieving information is closely similar to SQL. However, query languages for NoSQL databases are mostly specific to the particular implementation of the database. For example, CQL can be used only with Apache Cassandra NoSQL database, which has been used for the prototype implementation of Social BIMCloud. However, research is ongoing for standardizing NoSQL query languages (Grolinger et al.

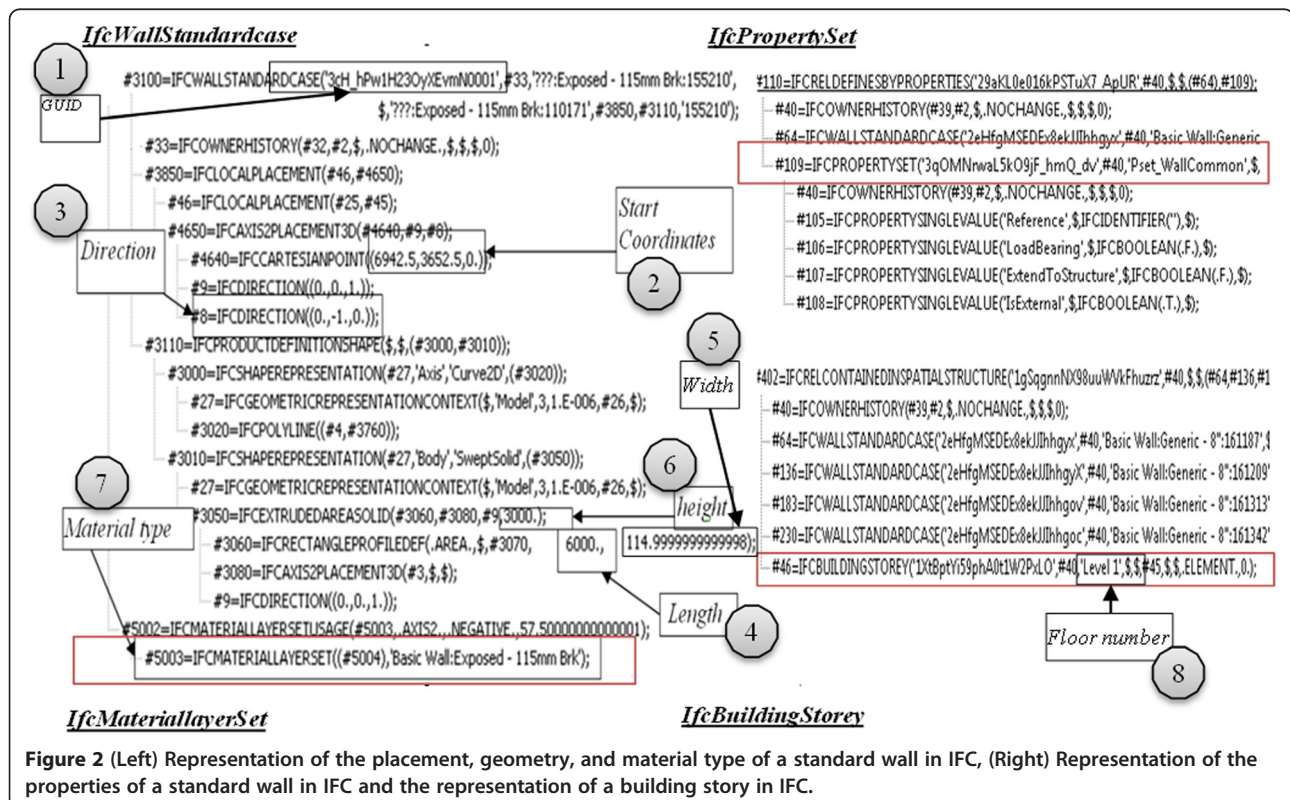
2013; Bach and Werner 2014). Therefore, in the future, a unified and standardized query language can be potentially expected to be used for Social BIMCloud. The data schema of Social BIMCloud may be published through ontology based web service standards like SAWSDL (W3C 2007). SAWSDL is an XML based interface definition language that is used for describing the functionality offered by a web application. SAWSDL provides a machine readable description of how a web application can be called and the ontology of the parameters that it expects. By exposing its schema to the end users, Social BIMCloud can facilitate customized queries and schema updating. Query and schema updates will however be modulated according to the access rights of the end user attempting to perform an operation.

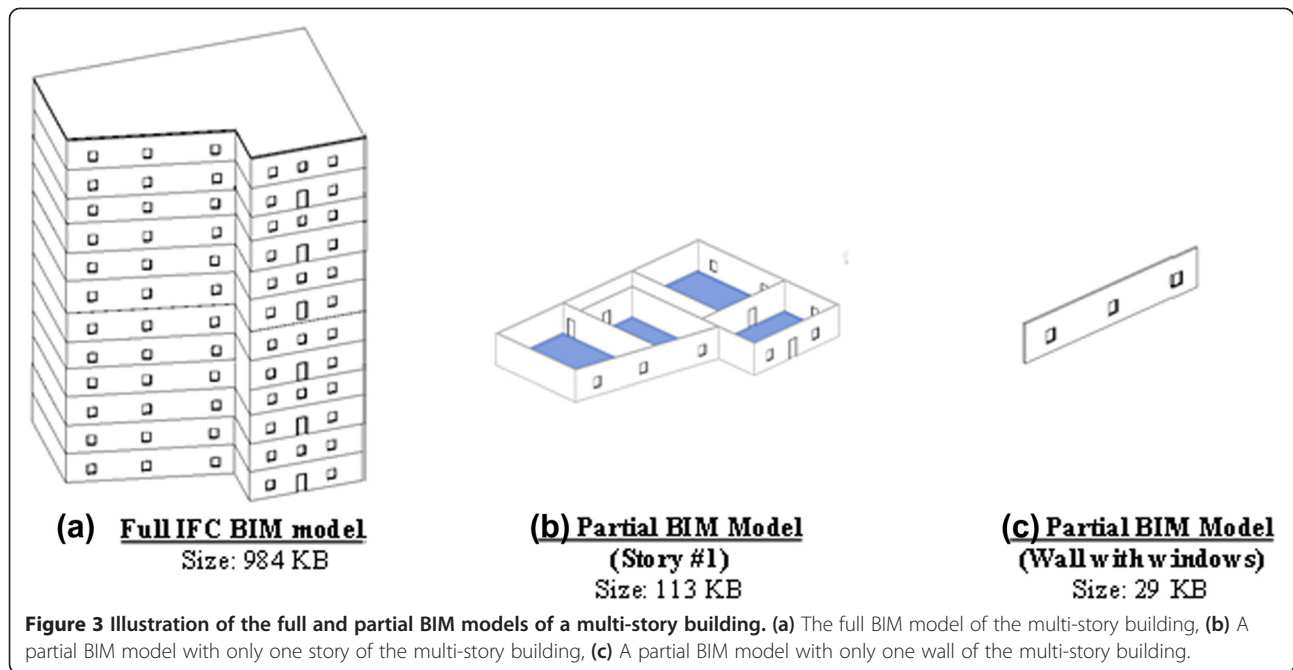
Data upload and extraction layer

This layer interprets the data received from end users through the data capture and flow controller layer and modifies the dataset into a database compatible format (the NoSQL database used in the Social BIMCloud). For example, when an open BIM standard model is inputted to the Social BIMCloud, program modules in the data upload and extraction layer parses the BIM model and

extracts the key building information like building geometry, location and orientation of building elements, relation among the building elements, material, and schedule. The extracted key information is then uploaded to the distributed database in the data storage layer by program modules through database queries. Similarly, the data upload and extraction layer also facilitates the extraction of building information from the database and writing to BIM files in open standard. To parse open BIM standard based files, the parser program can be written in standard programming languages like Java and C# by using open source libraries (for example, JSDAI for IFC). Figure 2 shows a standard building wall represented in the IFC schema. The data structure of IFC comprises entities containing information fashioned in a tree-like hierarchy. To represent a standard wall in IFC, a collection of sub-entities under four super entities, namely *IfcWallStandardCase*, *IfcMaterialLayerSet*, *IfcPropertySet*, and *IfcBuildingStorey*, are required to represent the building geometry, material, properties of individual building elements, and story information, respectively. In addition, other relational IFC entities like *IfcRelDefinesByProperties* are required to represent the relations among these four entities. Although an IFC file of a large building may be huge in size, the key values stored in the IFC entities required for representing the

building are much smaller in size. For example, as shown in Figure 2, the key entities required for representing a standard wall in IFC are - (1) GUID of the wall, (2) placement coordinates of the wall, (3) direction of the wall, (4) length, (5) width, (6) height, (7) material type, and (8) floor number. The GUID is the key identifier of a building element which remains consistent through import and export via BIM software like Autodesk Revit. The GUID is used in this framework for identifying the other information like length, width, and material to a particular building element. Therefore, this framework can perform the exchange of partial building models, i.e., the exchange of a particular building element or a group of building elements identified through the GUID. Similar to IFC the other open BIM standard, for example, gbXML also contains key information represented in a pre-defined schema. gbXML uses the XML schema to represent building elements. Therefore, APIs for such open BIM standards can be used to automatically extract and exchange only the key information instead of full building model files among project partners. Figure 3 shows the variation in the sizes of full and partial models of a simple multi-story building model. This example shows that a reduction in size of around 11% to 97% or more can be achieved by splitting the original model into partial models. The data upload and





extraction layer of Social BIMCloud allows the end users to download or upload partial building model by dynamically splitting or merging it from or to the full building model stored on Social BIMCloud server. Smaller files can transfer over the Internet more efficiently and therefore Social BIMCloud improves the data transfer efficiency. Moreover, unlike existing cloud-based BIM frameworks, Social BIMCloud allows end users to merge new building or non-building related information to the common integrated BIM model stored on Social BIMCloud through partial merging without disturbing the original model. Through this approach, Social BIMCloud provides all the users with the access to new additions or modifications to the common BIM model instantly.

This layer also captures ongoing formal social interactions in a construction project like change orders and RFIs. A change order is usually issued by a project manager as approval of any change required to the original building design. An RFI is issued by a contractor when he is seeking additional information on a construction or approval for any deviation in construction. Numerous formal interactions take place through RFIs and change orders, and informal interactions are common throughout the construction phase of a building. Documenting and managing such interactions would result in the creation of a knowledge base which could be used to improve the supply chain during the project and later through value engineering. Table 1 shows examples of the type of social information that should be hosted in the integrated BIM server of the Social BIMCloud framework. Table 1 shows the type of attributes, examples of their values, and the mandatory attributes

required for capturing a social interaction. Table 1 also lists the method of data capture used by Social BIMCloud for these various attributes. Every social interaction that takes place in a construction project is associated with one or more building elements (for example, wall, floor, and spaces). The Social BIMCloud framework also allows the non-technical end users to make informal social interactions through comments like “good,” “bad,” “needs to be redone,” “change tile,” and “dissatisfactory construction” on the building components. However, end users (like owner and subcontractors) who are involved in the exchange of social interactions may not be familiar with BIM standards like IFC, and therefore require a user friendly and visual medium. Therefore the Social BIMCloud is integrated with standard BIM software. BIM software like Autodesk Revit provides open source API which allows application developers to integrate their external applications with customized plugin. The Revit plugin for Social BIMCloud performs two functions – (1) uploading social interactions from Revit to the Social BIMCloud platform and (2) downloading social interactions from the cloud platform and displaying it in Revit. With this plugin, users can easily view others’ comments and upload their own ones by simply selecting a building model in Revit and clicking the “upload comments” dialog box, without the need for prior knowledge with Revit. Web pages can also be developed to connect and interact with the social information stored in Social BIMCloud. A detailed discussion on information update and retrieval through this plugin is presented in Section Information updating and retrieval.

Data storage layer

The data storage layer of Social BIMCloud consists of a distributed column-based NoSQL database. The key information extracted from partial or full open BIM standard or other input sources like BIM software or web pages is stored in this database. NoSQL databases have the DDL (data definition language) and DML (data manipulation language) constructs similar to the relational databases, but are designed specifically to handle large volumes of data. Unlike in relational databases, Social BIMCloud server can handle dynamically added columns with different data types in a column family (the concept of a column family is similar to tables in relational database). This property of NoSQL databases can facilitate a server based BIM with flexible data schema. Data required new domains or functions can be easily added without altering the relations between the entities in the original data schema. The Social BIMCloud server also facilitates data recovery and improved query speed through data replication, and horizontal data partitioning. Therefore large amount of BIM data can be exchanged or complex queries can be executed at a fast speed. Also, the NoSQL database can be deployed on a distributed platform that comprises of many cloud-based or non-cloud-based computers. The architecture of the Social BIMCloud server is shown in Figure 4 (left). The Social BIMCloud server comprises multiple nodes (computers), each hosted on a cloud server (may be located in different regions around the world) and performs in

parallel with each other. The number of these nodes can be increased or decreased according to the size of the project. Data inserted into this server through any node is equally distributed among all the nodes. In a construction project, the resources (cloud machines) can be increased or decreased according to the requirement of speed and storage of the project without having to make unnecessary investments on hardware infrastructure.

The Social BIMCloud server facilitates automatic replication of data and therefore provides high data availability and disaster recovery. The number of times which the original data should be replicated is called the replication strategy and can be defined by end users. Figure 4 (left) shows a data set, 'Data 1' being inserted into the Social BIMCloud. In this case, a replication strategy equal to 3 is set. 'Data 1' is broken internally (called sharding) into three parts by the node through which the data is being inserted (node 8) and distributed across through an internal mechanism. The number of copies that each piece of data (called shards) has is equal to 3 (Figure 4 (left)) according to the replication strategy. Data sharding means storage of rows of one column family in different servers so that parallel processing can be performed while retrieving the complete set of data (Agrawal et al. 2011). Sharding improves query performance by facilitating selective querying in the Social BIMCloud server, where only a particular shard may be queried for retrieving selective results. Sharding also facilitates data recovery in the

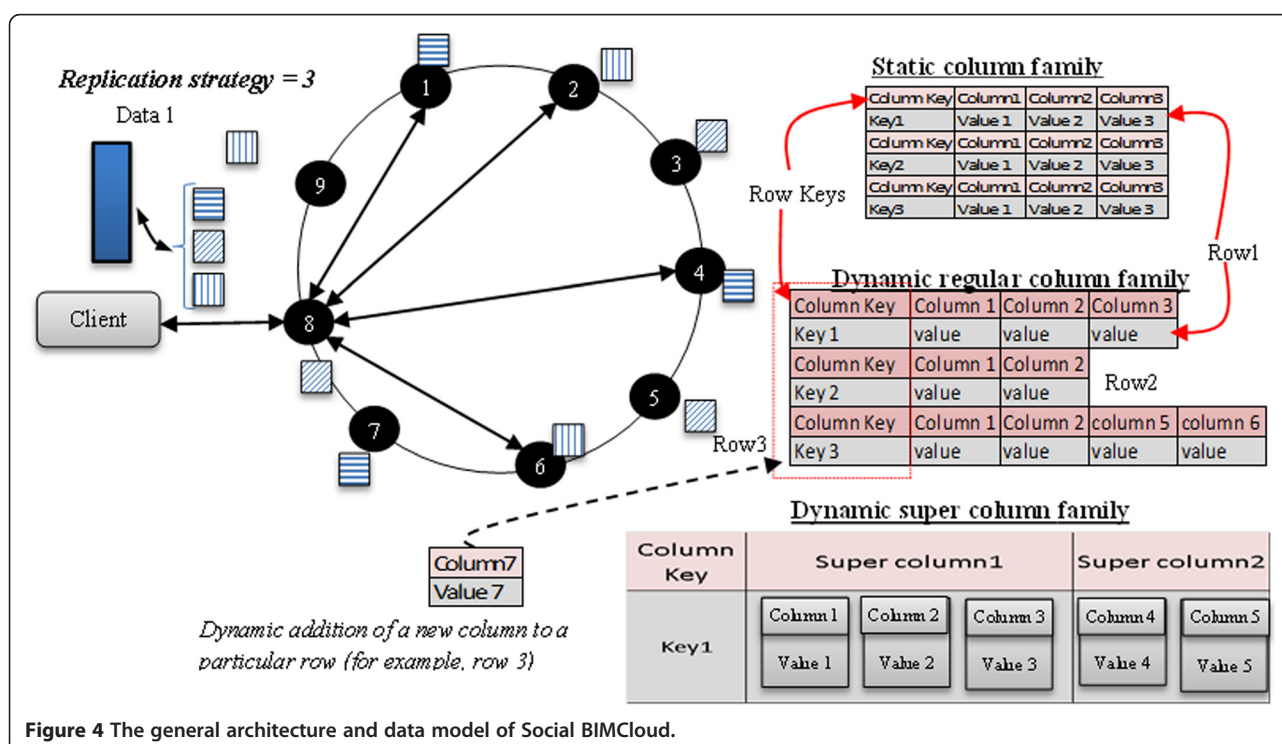
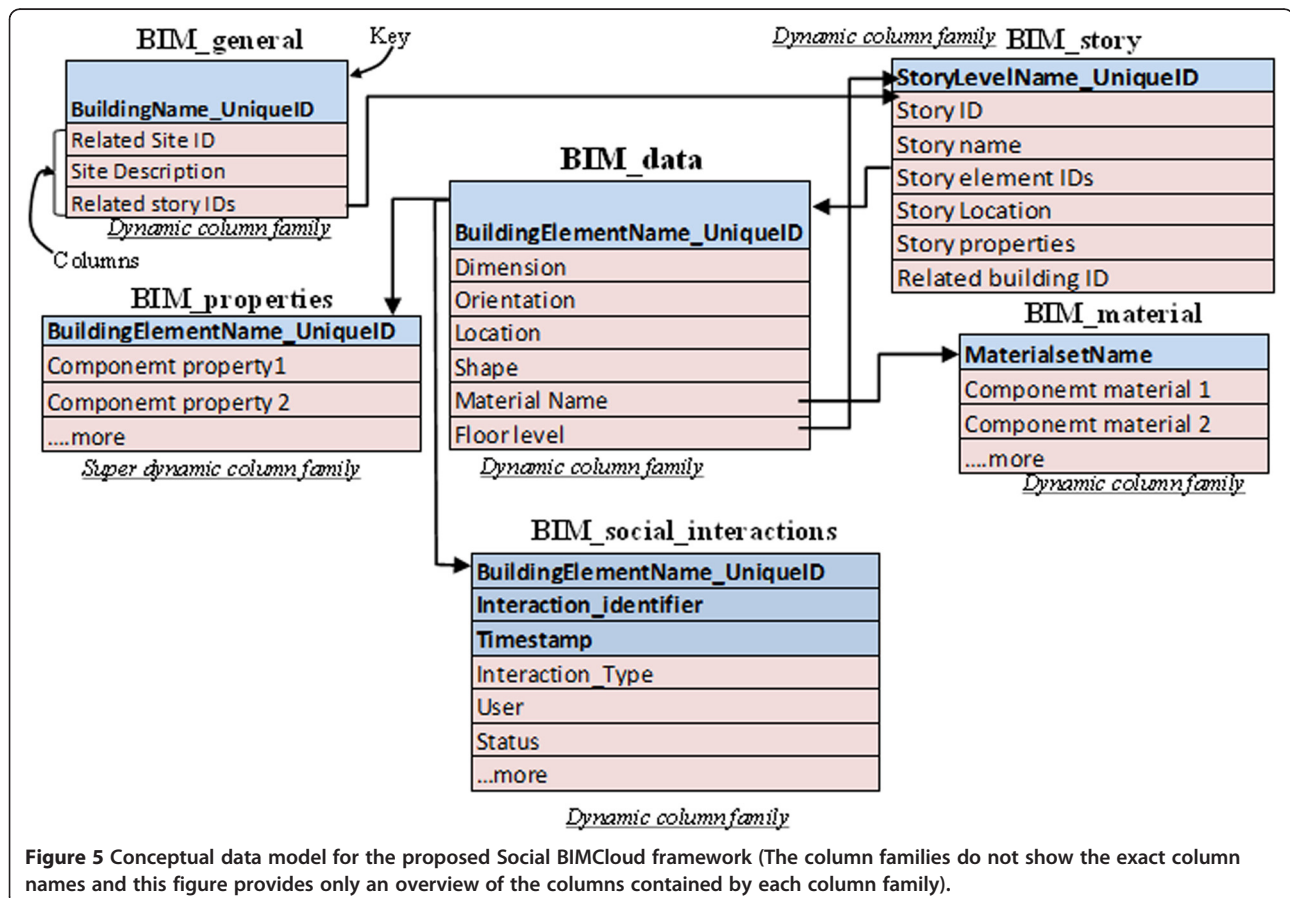


Figure 4 The general architecture and data model of Social BIMCloud.

case of failure of one or more nodes. For example, in case of failure of three nodes, say nodes 1, 2, and 3, the database can automatically retrieve “Data 1” through nodes 4, 5, and 6. Cloud computing (cloud service providers like Amazon EC2 web services) makes the concept of data replication and data partitioning on Social BIMCloud server significantly easier by providing virtually unlimited capacity on demand and taking care of all the necessary database administration tasks.

The data model of the Social BIMCloud server is based on a column oriented data model, the main concepts of which are column families and columns. A column family may contain columns of different data types like text, integers, and decimals. Each row in a column family comprises two parts – a unique key and a set of values. Both of these parts are basically columns designated for different purposes. As shown in Figure 4 (right), column key and the set of columns like column 1, column 2, and column 3, are all columns containing values retrieved from the user inputted dataset. The value in the column ‘Column Key’ (e.g. ‘Key 1’) is the key of each row (e.g. row 1). A row key should be unique in a column family and may consist of one or more columns. The row key is used as an identifier for creating

primary indexes on column families for faster data retrieval. A column family may be broadly of two types – (1) static and (2) dynamic. A static column family uses a relatively static set of column names and is similar to a relational database table. A dynamic column family allows end users to dynamically add arbitrary application and user supplied column names without having to conform to a particular fixed schema. For example, the dynamic column family shown in Figure 4 (right) has a different set of columns for each row. The dynamic columns may be subdivided into two groups – (1) regular and (2) super. A regular column is a tuple in the column family comprising a name and a value (for example ‘column 6’ and ‘value’ as shown in Figure 4 (right)). Super columns (for example, ‘Super column 1’ as shown in Figure 4 (right)) are columns within columns, and add levels of nesting to the structure of the regular column family. Such nested structures are useful for representing the tree-like structure of standard BIM, for example, the tree-like structure of building element properties, where a building element has several group of properties with many sub-layers. Figure 5 shows the proposed data model for the Social BIMCloud server. Figure 5 shows the row keys (in bold and



blue) and column names (normal and red) of each column family. The data model of Social BIMCloud comprises the following six column families:

- (1) *BIM_data* – *BIM_data* is a dynamic regular column family that stores the geometry, location, and orientation-related information of a building element. The row key of this column family is the building element name (e.g. *IfcWallStandardcase*) appended with a unique GUID of the respective building element extracted from the BIM file in open standard. Each building element data is stored in a separate row with its own unique key in order to facilitate retrieval of partial information, for example the geometry of one particular wall. *BIM_data* also stores a reference to the material type, story name, element properties, the ID of the building to which the building element belongs to, and social interactions related to the building element that has taken place during the construction process to the column families, *BIM_material*, *BIM_story*, *BIM_properties*, *BIM_general*, and *BIM_social_interactions* respectively. These references are used to extract partial information like material type of a building element or social interactions related to it.
- (2) *BIM_general* – This column family stores all the information other than that of the building itself. For example, project name, owner name, site ID and site description are stored. The row key of this column family is the building name (for example, “Two Story Building”) appended with the common GUID of the building. *BIM_general* stores a reference *BIM_story* through story ID. With this data model, the whole building model may be extracted by using information like building name, project name, or site name. *BIM_general* has been designed as a dynamic regular column family as the data stored in it do not have a complex hierarchical structure.
- (3) *BIM_story* – This column family contains information related to a building story like story name or number, story ID, story placement, and properties of a building story. *BIM_story* has been designed as a dynamic regular column family as the data structure related to a story is not complex. The information related to each story is stored in different rows each with a unique row key. The row key of this column family is the story name (for example, “level 2”) appended with the GUID of the respective story. *BIM_story* has reference to building element ID of the *BIM_data* column family in order to facilitate retrieval of individual stories (containing its building elements) through one query.
- (4) *BIM_properties* – This column family stores properties of a building element, story, or material like type, use, and constraints. A building element may have several levels of properties and sub-properties. Therefore, in order to store the property data in a hierarchical structure, it has been designed as a super dynamic column family. The row key of this column family is building name appended with the GUID of the building element. Therefore, in this column family, each row contains the properties related to individual building elements. This facilitates faster information retrieval of the complete BIM information building element with its geometry and properties.
- (5) *BIM_material* – This column family stores material related information like material name and material type. It contains information of single layer and multi-layer materials like sub-material name and thickness. The row keys of this column family are unique material names. As discussed earlier, *BIM_data* stores material name and refers to *BIM_material* through the unique material name for information on materials.
- (6) *BIM_social_interactions* – This column family captures the information (as described in Table 1) on social interactions that take place during a construction project. The row key of *BIM_social_interactions* comprises three columns, namely building element GUID, social interaction ID, and timestamp. It has been done so that the building element ID is not unique in this column family, as one building element may contain more than one social interaction. However, the building element ID is the reference of this column family from *BIM_data*, and therefore should be indexed for faster queries. The data model of this column family is designed to create primary indexes on all the three key columns for data sorting according to any of them.

Figure 6 shows the example data model developed based on the IFC BIM standard and the information model for social interactions described in Table 1. Figure 6 also shows the examples of attributes stored by each column family, the relation between column families, and the sequence of extraction of a partial building element.

Scalability of the Social BIMCloud framework

Scalability and flexibility are the key features of cloud computing. The Social BIMCloud framework is scalable in terms of performance and functionality. The cloud instances of the Social BIMCloud framework (as shown in Figure 1) can be increased or decreased to maintain a standard performance depending upon the size of the BIM model stored in the Social BIMCloud and the

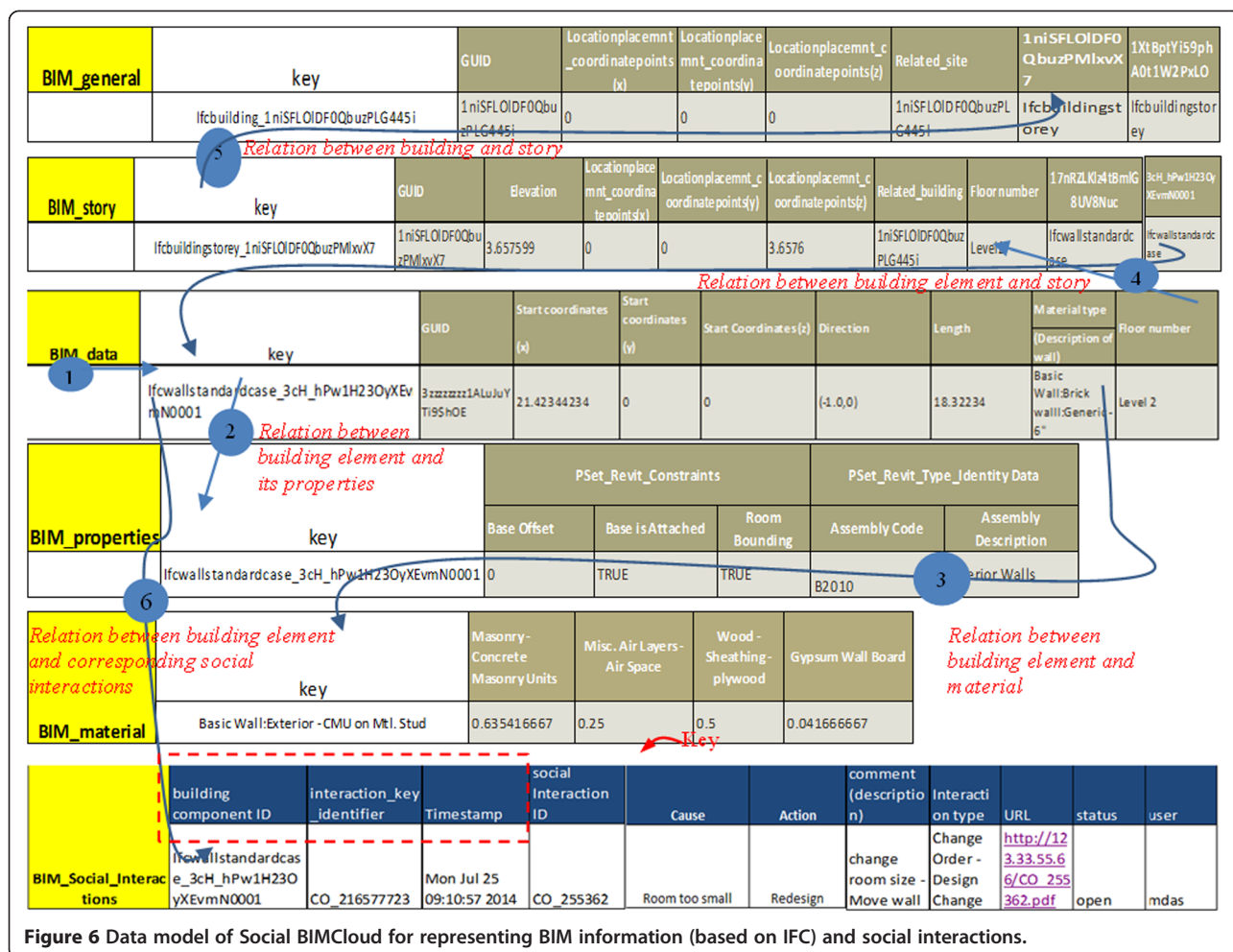


Figure 6 Data model of Social BIMCloud for representing BIM information (based on IFC) and social interactions.

number of operations being performed on it. For example, during the construction phase, the number of queries for data extraction and updating on the Social BIMCloud server would be more than that in the maintenance phase. During this period, new cloud instances could be launched in order to facilitate parallel processing and hence increase the speed of data retrieval and updating. The functions provided by the Social BIMCloud framework can be flexibly extended by integrating external applications like cost analysis and energy simulation software through APIs provided by the respective software. For example, the energy simulation software EnergyPlus (Crawley et al. 2001) provides an API for integrating the simulation engine with external applications. Therefore, the web service based framework for energy simulation presented in (Cheng and Das 2014) can integrate BIM models with the EnergyPlus simulation engine, which is deployed as web services. Energy simulation and BIM model parsing and updating are performed in the framework through various web services, which are connected and orchestrated using standardized web service technologies SOAP and BPEL. In

the Social BIMCloud framework, the data upload and extraction layer contains program modules for extracting information from the distributed cloud-based server. Cloud platforms like Amazon EC2 provides pre-configured instances like Elastic Beanstalk (EBS) for easy deployment of the web services though upload of the WAR (Web application Archive) file of the web service. The WAR file contains the web service program and its resources like files and database connections required by the web service in one aggregated file. WAR files can be easily created through software like Eclipse Web Tools Platform (Eclipse Foundation 2014). The scalability of Social BIMCloud will be discussed in the implementation part of the Social BIMCloud framework on Amazon web services in Section Extending the Social BIMCloud server.

Data security on the Social BIMCloud framework

The Social BIMCloud framework facilitates security of data through ACL (access control lists) on the distributed database, server side data encryption, and strong passwords. The ACL contains the list of user groups and

the corresponding access rights (e.g. read, write, delete, and revert), as decided by the administrator. These access rights can be granted at table and column level (based upon the key) of the distributed database. Therefore, different groups of users can be given access rights to different parts of a BIM model. For example, a contractor can be given read and write access to level 2 (level 2 is shown in Figure 6) by allowing write access to column with key 'Ifcbuildingstorey_1niSFLOIDF0QbuzPMIvX7'.

As shown in Figure 6, access can be granted to the same user group on the building elements and materials related to level 2 by querying the column families, finding the related columns and allowing access. Access rights are always assigned to a user by placing him or her in a group in the Social BIMCloud framework. If there is only one user pertaining to a particular access right, he or she is still assigned to a group. This is done to keep the management ACL limited to a few groups, instead of having to manage many individual users. Figure 7 shows an example of how access rights for writing information to Social BIMCloud may be distributed among the project stakeholders. Access rights may be of two types – (1) exclusive rights and (2) temporary rights for redesign. In exclusive rights, the owner has the right to create model partitions and distribute different parts of the building to different

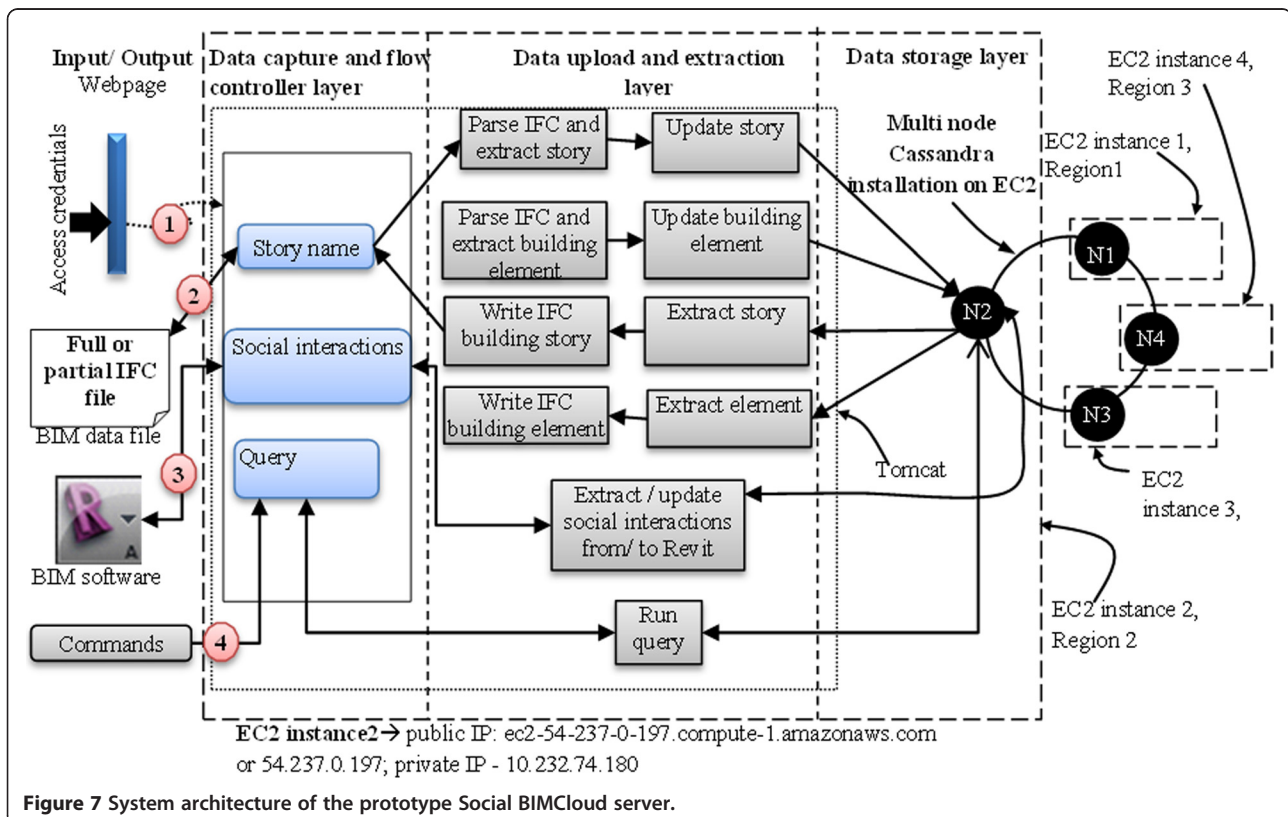
stakeholders (as shown in Figure 7(a)), who would have exclusive writing permissions. Similarly, only the architect and the designer (on the basis of owner's decision) are given rights to access a specific part of the building during the period of redesign.

The Social BIMCloud is a cloud-based framework and the data uploaded to it is eventually stored on the disks of a datacenter. Therefore, server side data encryption is deployed on the cloud instances in order to encrypt the data stored on the disks. The server side encryption is facilitated through a policy document which contains a list of the cloud instances that should be encrypted. A policy document is deployed when a new cloud instance is added to the Social BIMCloud framework. The admin can set customized password policies for users. This means that end users can change their password but would have to follow a particular format. In this way, strong alphanumeric passwords can be made mandatory for end users.

The prototype Social BIMCloud

Initial setup and connection for the Social BIMCloud

For demonstrative and testing purpose, the Social BIMCloud server is deployed with open standards like PHP (scripting language), Tomcat (web server), and Apache Cassandra (Column family-based NoSQL database). It is



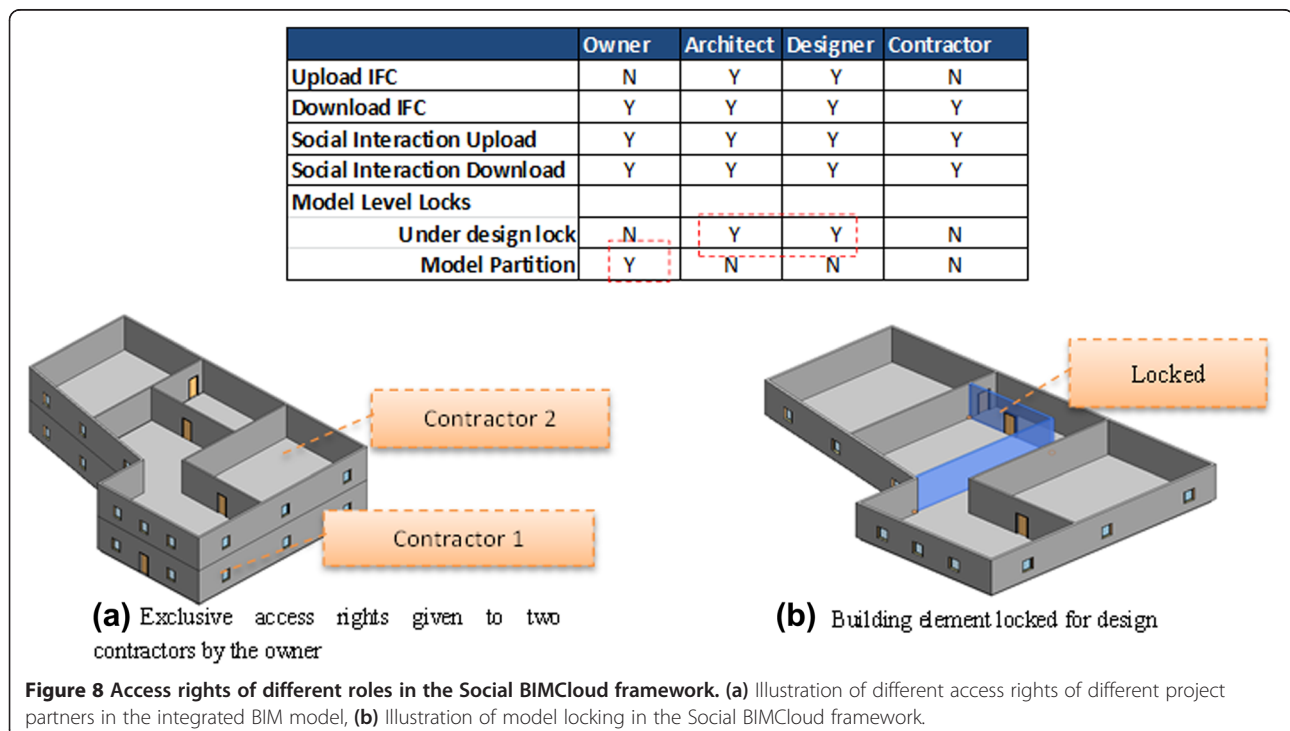
hosted by the cloud service provider, Amazon Web Services (AWS). Amazon Elastic Compute Cloud (EC2) is an IaaS web service provided by AWS, which provides cloud machines on demand. The Social BIMCloud server can be connected via SSH (secure shell) through a front end deployed in one of the instances or through a command line, by using the IP address of the EC2 instance.

Information updating and retrieval

As shown in Figure 8, the prototype Social BIMCloud server can facilitate partial upload and retrieval of IFC building files. End users can upload or retrieve full or partial IFC building models to or from the prototype Social BIMCloud server. End users have four options for upload and retrieval – (1) full IFC model, (2) one IFC building story, (3) one or more building elements, or (4) customized queries. For example, to download a building element, such as a wall, from the BIM model stored in the Social BIMCloud server, the PHP page 'Building element' accepts the unique ID of that wall, which can be easily viewed with any standard BIM software and a 'get' command on the PHP front page. This PHP module passes the control 'extract element' Java function in the data upload and extraction layer and executes a series of get commands on the N2 node of the Social BIMCloud server (Figure 8). As shown in (1) of Figure 6, the first command is executed on the column family *BIM_data* with the key being a combination of the unique ID and standard wall name. As shown in (2), (3), (4), and (5) of

Figure 6, get commands are executed on the column families *BIM_properties*, *BIM_material*, *BIM_story*, and *BIM_general* to get the key values of associated properties, associated material, story, and building (site and building as a whole) related information, respectively. The extracted key information is passed to the 'write IFC file' Java function in the same layer. An IFC file is written which contains only one wall and is downloaded through the front end. Similarly, for partial updating of an IFC file with the partial building model, for example, a new wall is provided. The building element PHP page with a put command similarly passes the control to the next layer. The parse 'IFC and extract building element' programs parse the IFC files and then pass the control to the 'update building element' program which executes put commands on the Cassandra instance and updates the key values in a similar manner as that described for extracting information in Figure 6.

The Social BIMCloud uses a visual interface developed with Autodesk Revit API to capture social interactions. This API provides library classes from which building information can be extracted and updated to a Revit model. Using VB.Net programming language, clients for extracting Revit building information and connecting to cloud instances can be created (Autodesk 2014). Figure 9 shows the prototype Revit plugin for connecting to cloud instances of the Social BIMCloud. As discussed in Section Data upload and extraction layer, the two operations provided by this plugin are – (1) uploading social interactions from Revit to the Social BIM platform and



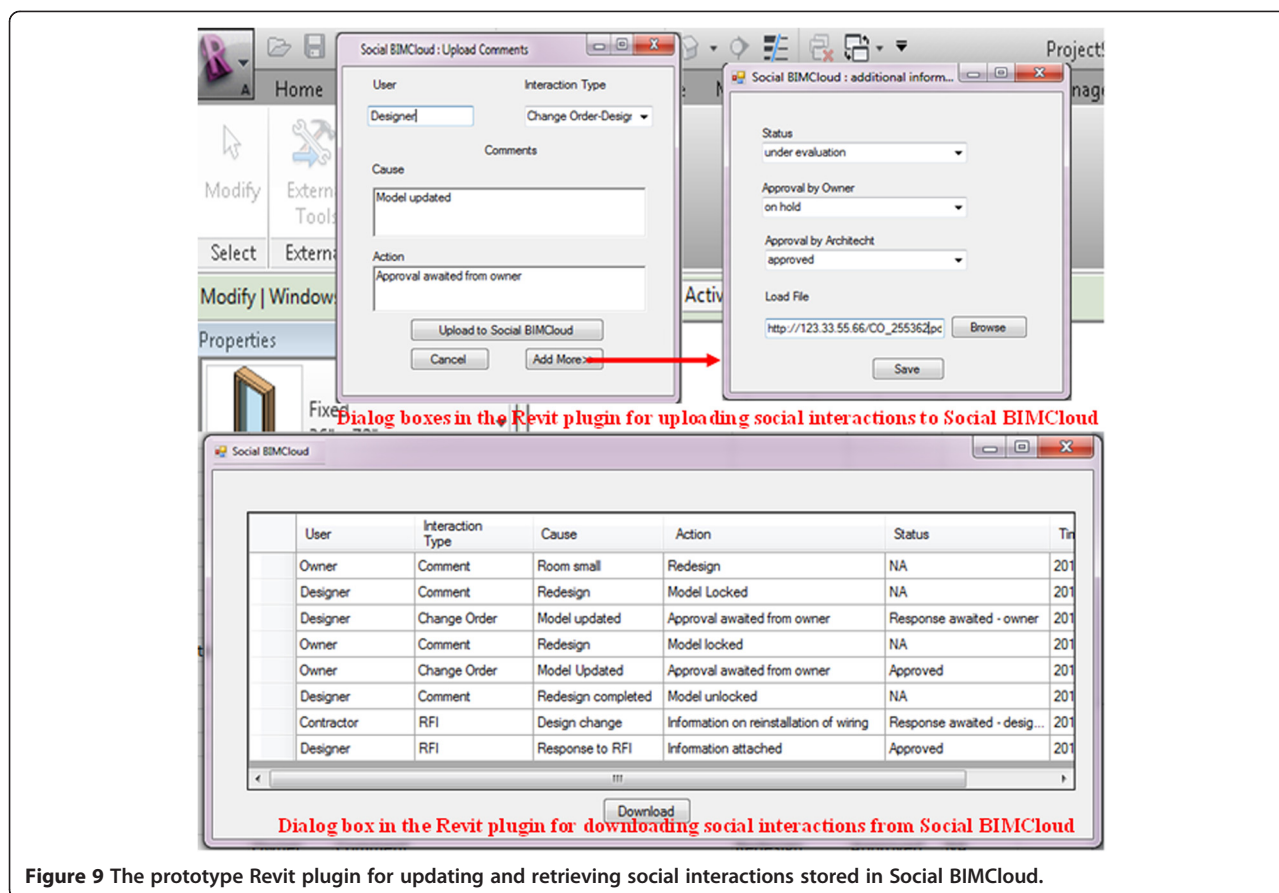


Figure 9 The prototype Revit plugin for updating and retrieving social interactions stored in Social BIMCloud.

(2) downloading social interactions from the Social BIM platform and displaying it in Revit. The Revit plugin uses customized dialog boxes for performing these two functions (as shown in Figure 9). For uploading social interactions to the Social BIMCloud, the end users select one or more building elements. In Figure 9, for example, an end user selects the three windows (selection is highlighted) and runs the Revit plugin under the External Tools tab. End users can add more information to this change order by clicking on the “Add more” button, and entering information through the “additional information” dialog box. The Revit plugin also facilitates downloading of social interactions (for example, comments, change orders, and RFIs) associated with one or a particular group of building elements as shown in Figure 9. Figure 9 shows that the Social BIMCloud framework can facilitate communication among the project partners through the BIM model for faster decision-making, which will be explained in detail in Section Example scenario 2: social interactions for decision making.

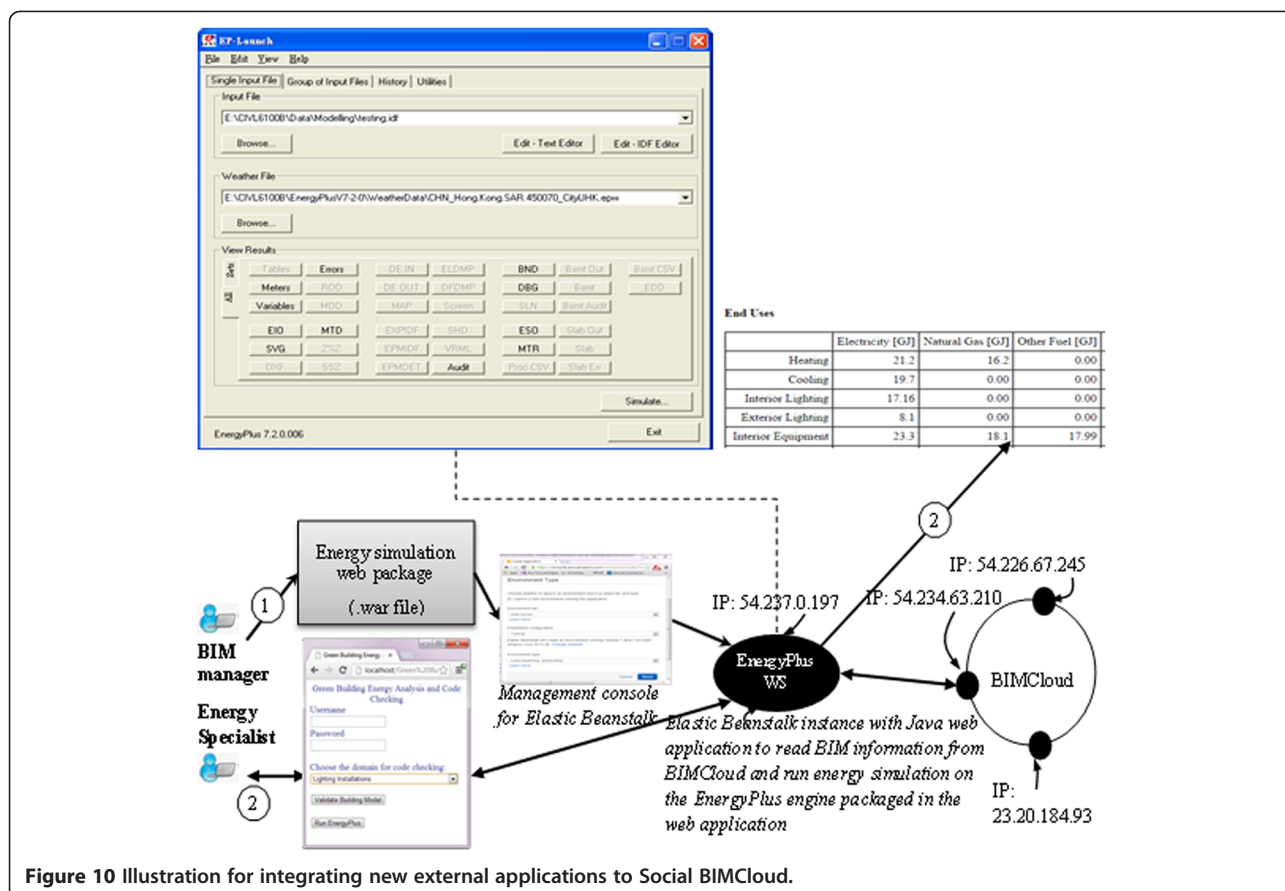
Extending the Social BIMCloud server

The Social BIMCloud server can be extended for functionalities by integrating external applications like services for cost estimation, scheduling, and energy

simulation (as discussed in Section Scalability of the Social BIMCloud framework). As shown in step 1 of Figure 10, the energy simulation web service (WAR file) is uploaded to an Amazon instance (Elastic Bean-Stalk or EBS) through a standard management console. In this case, the data model of Social BIMCloud may have to be extended to add energy related data. Energy related data may be stored in a new column family, say *BIM_Energy* or by altering *BIM_material* for including energy related information on materials. As shown in the step 2 of Figure 10, an energy specialist or a designer connects (through communication protocols like SSH) into the customized EBS instance to execute the energy simulation web service. The result of this execution is an EnergyPlus report file. EnergyPlus generates reports in both textual and graphical (graphs and charts) format. Figure 10 shows a part of an EnergyPlus report in textual format.

Results and discussion

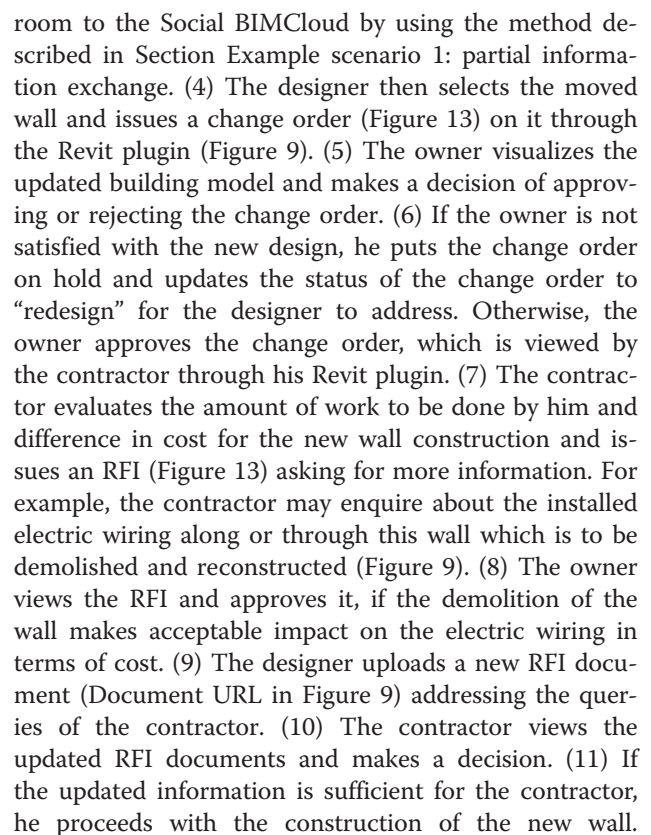
In this section, we present two example scenarios demonstrating the two main features of the Social BIMCloud framework – (1) partial information exchange and (2) decision making through social interactions.

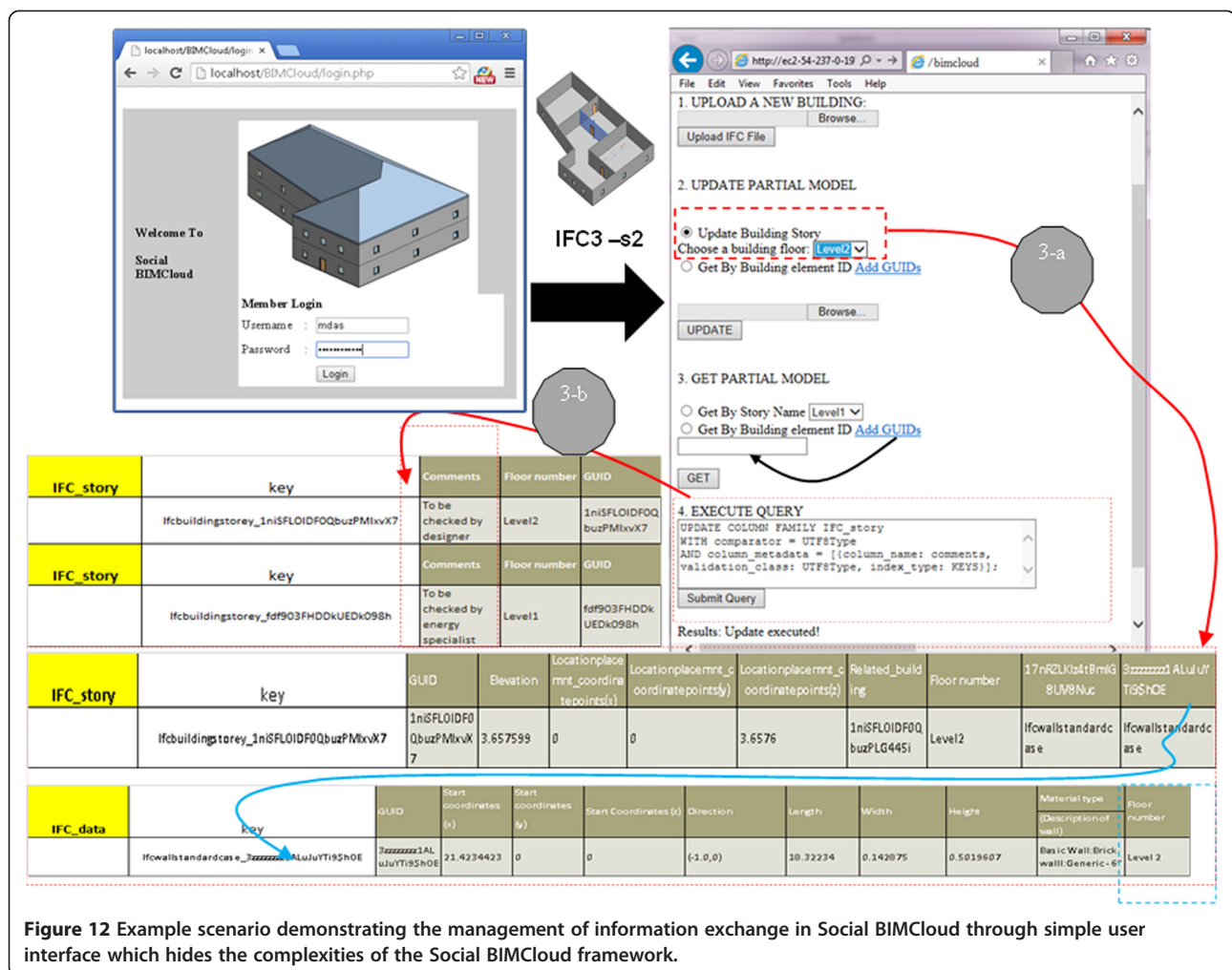


Example scenario 1: partial information exchange

Let us consider the design phase of a residential building in a high class and customer oriented development area. This BIM model is loaded to the Social BIMCloud by the designer (as shown by arrow 1 in Figure 11). The Social BIMCloud is accessible to end users via a webpage which authenticates the end users and opens the next web page (as shown in Figure 12). This web page performs four functions – (i) uploading a new building, (ii) downloading a partial model, (iii) uploading a partial model, and (iv) querying the Social BIMCloud using the CQL language provided by Cassandra. This scenario is about three project partners – architect, designer, and energy specialist who work on the same building model in parallel. The main architect downloads the level 1 (IFC2-s1) and level2 (IFC2-s2) of the IFC BIM model (arrow 2 of Figure 11), separately by consecutively selecting 'level 1' and 'level 2' and using the 'get' button in the 'get partial model' of the front end (Figure 12). The architect works on level 2 (IFC2-s2) first and changes the plan of level 2 (IFC3-s2) by removing the L-shaped wall and adding a straight wall (blue walls in Figure 11). The architect uploads level 2 (IFC3-s2) back to the Social BIMCloud. As shown in 3-a of Figure 12, a new wall is added to level 2. Also as shown in 3-b of

Figure 12, the architect updates the schema of the Social BIMCloud to accommodate comments and by executing an update query. He then notifies the designer (for example, the structural engineer or HVAC designer) and the energy specialist to perform checking on the changed design of the level that has been marked. The designer can download the full building model (arrow 5 of Figure 11) and can download the level on which comments with particular keywords have been made (in a case where multiple stories are involved). Meanwhile the architect is working on the partial IFC building model of level 1, which upon completion will be passed to the designer for review. The architect and designer can easily work on two partial parts of the same building and merge them back to the main Social BIMCloud when done. In level 1 (IFC2-s1), the architect removes the L-shaped wall and adds windows to promote cross-ventilation, so as to reduce the air conditioning load (IFC3-s1). He uploads the changed plan of level 1 (IFC3-s1) to the Social BIMCloud and notifies the energy specialist and designer. The energy specialist then performs simulation on level 1 (as discussed in Section Extending the Social BIMCloud server) by connecting to the Social BIMCloud (arrow 5 of Figure 11). In this scenario, we have seen that three



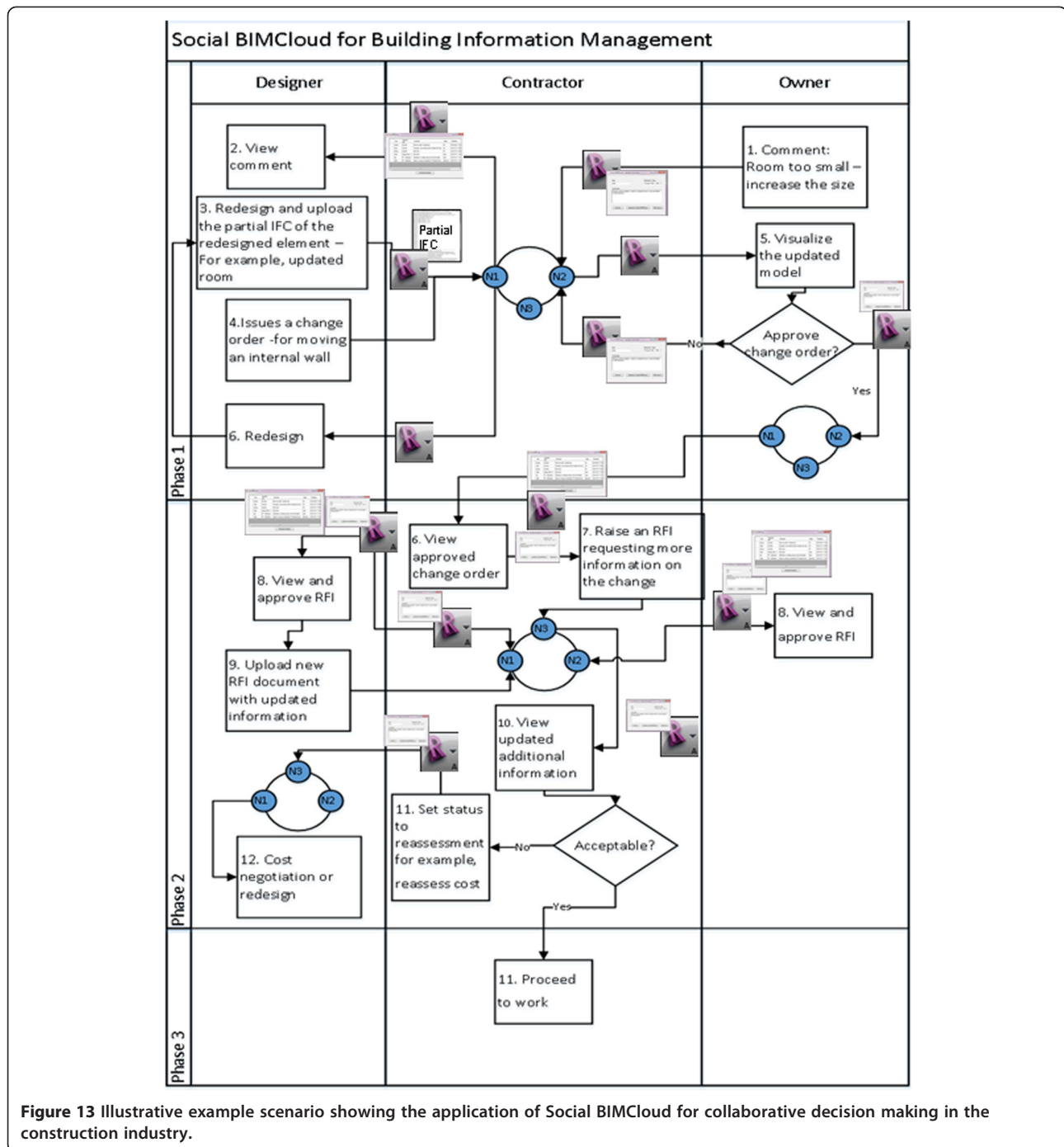


Otherwise, he puts the RFI on hold and sets to status to for example, “reassess cost”. (12) The designer may negotiate with the contractor or redesign the building as suitable.

Benefits and limitations

The Social BIMCloud framework facilitates lightweight and effective building information exchange for a fragmented construction industry and provides many benefits to information exchange in the AEC industry. First, Social BIMCloud is equally beneficial to large and small-sized BIM models as it facilitates dynamic splitting and merging of BIM information. Project partners can extract and upload partial building models (for example, one story or one room) from the integrated BIM model stored in the Social BIMCloud, through pre-defined or customized queries. Due to partial exchange, the size of data files being exchanged is reduced. This enables quick transfer of data, given the constraints of limited bandwidth at remotely located construction sites. Second, the data model schema in Social BIMCloud is based on but

not limited to the open BIM standard, IFC. Therefore, the data model of Social BIMCloud can be extended to include information from different AEC domains and aspects of building lifecycle, such as structural analysis data, energy analysis data, cost information and safety information. Third, Social BIMCloud provides a method for capturing and managing social interactions in the AEC industry through a visual tool for efficient decision making using a user friendly platform which has access to the most updated BIM information. Fourth, the framework is scalable in the way that storage and computing resources can be conveniently added to increase performance or removed to save cost, depending on the size of BIM models and the project requirements. Fifth, the Social BIMCloud framework implements an automatic data replication and data partitioning strategy on the BIM information stored on a distributed database management system. Data replication makes the Social BIMCloud framework fault tolerant. If any single node of the Social BIMCloud framework fails, the data lost can be automatically regenerated from the backup



replicated copies stored on the other nodes of the framework. Data partitioning facilitates division of the data stored in large BIM files into multiple parts and storage on different machines (nodes). This allows parallel reads and writes on the same building file, thereby improving performance. Sixth, Social BIMCloud also harnesses the inherent benefits of cloud computing, which utilizes a pay-per-use system of pricing for resources and high computing power. Finally, Social BIMCloud is

implemented as an IaaS (Information as a Service) which facilitates easy plug-and-play connectivity to external cloud-based or web-based applications, thereby making the Social BIMCloud framework highly extensible. The Social BIMCloud framework can also contain instances owned by different organizations or people and therefore create an integrated pool of BIM building information. However, on the downside, Social BIMCloud has three major limitations. First, the

performance of Social BIMCloud depends on the Internet bandwidth and the reliability of the cloud service providers. Therefore, Social BIMCloud should be implemented with cloud providers that have an efficient storage system of Social BIMCloud instances. Second, Social BIMCloud involves issues of ownership of data as different parties are involved in a construction project. Lastly, being based on a cloud platform, it may not be straightforward to move the entire system from one cloud platform provider to another, if needed, as the system configurations of two different service providers may be different.

Conclusion

Open BIM standards like IFC have been developed to facilitate interoperability and data consistency in the AEC industry, but due to the lack of a platform to facilitate object-based information exchange, information is still exchanged among project partners as BIM files. In this paper, we present a cloud-based BIM framework, namely Social BIMCloud, in order to facilitate object-based partial information exchange in an interoperable manner using open BIM standards. A methodology for storage and management of BIM information on a cloud-based framework has been presented in this paper. To accomplish this, a data model for NoSQL databases has been designed which can be hosted on cloud platforms. This NoSQL data model is capable of storing complex BIM information building element geometry, material information, and relationships with other building elements. The Social BIMCloud framework also captures and manages formal and informal interactions that take place among project partners during a construction project. For example, in the process of making changes to the building design during the construction phase, interactions take place among project partners through the process of information gathering and executing the task. The current approaches do not facilitate the integration of a BIM model with social interactions which are actually responsible for leading to changes in a BIM model. An integrated BIM model provides better visual aid in decision-making in an ongoing project. A methodology for developing visual tools for the capture and management of social interactions, starting from its data requirements to system implementation has been demonstrated in this paper through the integration of Social BIMCloud and Autodesk Revit. The data model of the Social BIMCloud is based upon but not limited to open BIM standards like IFC to facilitate interoperability by making the Social BIMCloud compatible with open BIM standards. In the future, other open BIM standards like gbXML and COBie can be explored and an integrated data model for the Social BIMCloud may be developed. Social BIMCloud may also be extended with an ontology based layer in

order to facilitate the publishing of its data schema through ontology based web service standards for customized queries and user-friendly schema integration.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

JC and MD developed the conceptual framework of Social BIMCloud for partial BIM and social interaction exchange. MD studied the latest technologies in cloud computing and BIM and developed the prototype for partial information exchange. SSK developed the prototype Revit plugin for integrating Social BIMCloud with standard BIM software (Autodesk Revit). All the three authors have worked together to integrate Social BIMCloud to be a platform for visualization and efficient information exchange. All authors read and approved the final manuscript.

Received: 6 November 2014 Accepted: 24 February 2015

Published online: 12 March 2015

References

- Agrawal, D, Abbadi, AE, Das, S, & Elmore, AJ. (2011). *Database scalability, elasticity, and autonomy in the cloud*. Paper presented at the Proceedings of the 16th international conference on Database systems for advanced applications - Volume Part I, Hong Kong, China.
- Autodesk. (2013). *Autodesk 360: work wherever you are – safely (white paper)* (Vol. 2013, Autodesk 360 Security Overview 2012 Vol. 25 December).
- Autodesk (2014). Cloud computing and your Autodesk product plugins. <http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=17136545>. Accessed 12 July 2014.
- Bach, M, & Werner, A. (2014). Standardization of NoSQL database languages. In S Kozielski, D Mrozek, P Kasprowski, B Malysiak-Mrozek, & D Kostrzewa (Eds.), *Beyond databases, architectures, and structures*. (Vol. 424, pp. 50–60, Communications in Computer and Information Science): Springer International Publishing.
- Beetz, J, Berlo, LV, Laat, RD, & Helm, PVD. (2010). BIMSERVER.ORG – an open source IFC model server. In *CIB W78 2010: 27th International Conference, Cairo, Egypt, 16–18 November 2010*.
- BIM9 (2014). www.bim9.com. Accessed 10 October 2014.
- BIMObject (2014). <http://bimobject.com/>. Accessed 10 October 2014.
- CaddForce (2014). <http://www.caddforce.com/>. Accessed 10 October 2014.
- Chan, S, & Leung, N. (2004). Prototype web-based construction project management system. *Journal of Construction Engineering and Management*, 130(6), 935–943. doi:10.1061/(asce)0733-9364(2004)130:6(935).
- Chen, P-H, Cui, L, Wan, C, Yang, Q, Ting, SK, & Tiong, RLK. (2005). Implementation of IFC-based web server for collaborative building design between architects and structural engineers. *Automation in Construction*, 14(1), 115–128. <http://dx.doi.org/10.1016/j.autcon.2004.08.013>.
- Cheng, JCP, & Das, M. (2014). A BIM-based web service framework for green building energy simulation and code checking. *ITcon*, 19, 150–168.
- Chuang, T-H, Lee, B-C, & Wu, I-C. (2011). Applying cloud computing technology to Bim visualization and manipulation. In *Proceedings of the 28th ISARC, Seoul, South Korea* (pp. 144–149).
- Crawley, DB, Lawrie, LK, Winkelmann, FC, Buhl, WF, Huang, YJ, Pedersen, CO, Strand, RK, Liesen, RJ, Fisher, DE, Witte, MJ, & Glazer, J. (2001). EnergyPlus: creating a new-generation building energy simulation program. *Energy and Buildings*, 33(4), 319–331.
- Dainty, ARJ, Briscoe, GH, & Millett, SJ. (2001). Subcontractor perspectives on supply chain alliances. *Construction Management and Economics*, 19(8), 841–848.
- Eclipse Foundation. (2014). *Eclipse - web tools platform*.
- Faraj, I, Alshaw, M, Aouad, G, Child, T, & Underwood, J. (2000). An industry foundation classes web-based collaborative construction computer environment: WISPER. *Automation in Construction*, 10(1), 79–99. [http://dx.doi.org/10.1016/S0926-5805\(99\)00038-2](http://dx.doi.org/10.1016/S0926-5805(99)00038-2).
- Fathi, F, S, Abedi, M, Rambat, S, Rawai, S, & Zakirudin, M. Z. (2012). Context-aware cloud computing for construction collaboration. *Journal of Cloud Computing*, 2012, 1–11.
- Graphisoft (2013). BIM Server. http://www.graphisoft.com/hk/bim_server/.
- Grilo, A, & Jardim-Goncalves, R. (2011). Challenging electronic procurement in the AEC sector: a BIM-based integrated perspective. *Automation in Construction*, 20(2), 107–114. <http://dx.doi.org/10.1016/j.autcon.2010.09.008>.

- Grolinger, K, Higashino, WA, Tiwari, A, & Capretz, MA. (2013). Data management in cloud environments: NoSQL and NewSQL data stores. *Journal of Cloud Computing: Advances, Systems and Applications*, 2, 22.
- Huberman, BA, & Adami, LA. (2004). Information dynamics in the networked world. In *Lecture notes in physics* (pp. 371–398). Berlin Heidelberg: Springer-Verlag.
- Javaja, P, Suwal, S, Porkka, J, Savisalo, A, & Kokko, P. (2012). *Social interaction in urban planning projects*. Paper presented at the CIB W78 2012: 29th International Conference, Beirut, 17–19 October.
- Jernigan, FE. (2008). *BIG BIM little bim* (2nd ed.). Site Press.
- Jiao, Y, Zhang, S, Li, Y, Wang, Y, & Yang, B. (2013). Towards cloud augmented reality for construction application by BIM and SNS integration. *Automation in Construction*, 33(0), 37–47. <http://dx.doi.org/10.1016/j.autcon.2012.09.018>.
- Kang, H, & Lee, G. (2009). Development of an object-relational IFC server. In *International Conference on Construction Engineering and Management/Project Management (ICCEM/CCPM)*, Jeju, Korea.
- Katranuschkov, P, Gehre, A, & Scherer, RJ. (2003). An ontology framework to access IFC model data. *ITcon*, 8(Special Issue eWork and eBusiness), 413–437.
- Kumar, B, Cheng, JCP, & McGibbney, L. (2010). Cloud computing and its implications for construction IT. In *International Conference on Computing in Civil and Building Engineering*, Nottingham, UK (pp. 315–324).
- Liu, R, Issa, RRA, & Olbina, S. (2010). Factors influencing the adoption of building information modeling in the AEC industry. In *International Conference in Computing in Civil and Building Engineering*, University of Nottingham, United Kingdom, 30 June - 2 July 2010. Nottingham, United Kingdom: Nottingham University Press.
- Mell, P, & Grance, T. (2011). *The NIST definition of cloud computing*. National Institute of Standards and Technology (NIST).
- Nawaz, S, Efstratiou, C, Mascolo, C, & Soga, K. (2012). *Social sensing in the field: challenges in detecting social interactions in construction sites*. Paper presented at the Proceedings of the 1st ACM workshop on Mobile systems for computational social science, Low Wood Bay, Lake District, UK.
- Nepal, M, Zhang, J, Webster, A, Staub-French, S, Pottinger, R, & Lawrence, M. (2009). Querying IFC-based building information models to support construction management functions. In *Construction Research Congress 2009* (pp. 506–515): American Society of Civil Engineers.
- Nour, MM. (2007). Manipulating IFC sub-models in collaborative teamwork environments. In *24th CIB W-78 Conference*, Maribor, Slovenia.
- Nour, MM, & Karl, B. (2008). An open platform for processing IFC model versions. *Journal of Tsinghua Science and Technology*, 13, 126–131.
- Plume, J, & Mitchell, J. (2007). Collaborative design using a shared IFC building model—learning from experience. *Automation in Construction*, 16(1), 28–36. <http://dx.doi.org/10.1016/j.autcon.2005.10.003>.
- Porwal, A, & Hewage, KN. (2013). Building Information Modeling (BIM) partnering framework for public construction projects. *Automation in Construction*, 31(0), 204–214. <http://dx.doi.org/10.1016/j.autcon.2012.12.004>.
- Redmond, A, & Smith, B. (2011). *Exchanging partial BIM information through a cloud-based service: testing the efficacy of a major innovation*. London: IBEA Conference, South Bank University.
- Sargent, K, Hyland, P, & Sawang, S. (2012). Factors influencing the adoption of information technology in a construction business. *Australasian Journal of Construction Economics and Building*, 12, 2. AJCEB.
- Shen, W, Shen, Q, & Sun, Q. (2012). Building Information Modeling-based user activity simulation and evaluation method for improving designer–user communications. *Automation in Construction*, 21(0), 148–160. <http://dx.doi.org/10.1016/j.autcon.2011.05.022>.
- Tanyer, AM, & Aouad, G. (2005). Moving beyond the fourth dimension with an IFC-based single project database. *Automation in Construction*, 14(1), 15–32. <http://dx.doi.org/10.1016/j.autcon.2004.06.002>.
- Tyler, J, Wilkinson, D, & Huberman, B. (2003). Email as spectroscopy: automated discovery of community structure within organizations. In M Huysman, E Wenger, & V Wulf (Eds.), *Communities and technologies* (pp. 81–96). Netherlands: Springer.
- W3C (2007). Semantic Annotation for WSDL and XML Schema. <http://www.w3.org/TR/sawSDL/>. Accessed 23 June 2014.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com